

– Lösung zur Praktikumsaufgabe 1 –

Thema: *Off-Line-Scheduling*

1. a) $H = 70$, $f \geq 7$, $f \in \{2, 5, 7, 10, 14, 35\}$, Ignorieren des 1. Frame Size Constraint liefert $f \in \{2, 5\}$

- b) Die in H zu planenden Jobs $J_{i,k}(t_{r,ik}, t_{e,ik}, t_{d,ik})$ sind in Tabelle 1 aufgeführt.

$J_{1,1}(0, 1, 5)$	$J_{1,2}(5, 1, 10)$	$J_{1,3}(10, 1, 15)$	$J_{1,4}(15, 1, 20)$	$J_{1,5}(20, 1, 25)$
$J_{1,6}(25, 1, 30)$	$J_{1,7}(30, 1, 35)$	$J_{1,8}(35, 1, 40)$	$J_{1,9}(40, 1, 45)$	$J_{1,10}(45, 1, 50)$
$J_{1,11}(50, 1, 55)$	$J_{1,12}(55, 1, 60)$	$J_{1,13}(60, 1, 65)$	$J_{1,14}(65, 1, 70)$	
$J_{2,1}(0, 1, 9)$	$J_{2,2}(7, 1, 16)$	$J_{2,3}(14, 1, 23)$	$J_{2,4}(21, 1, 30)$	$J_{2,5}(28, 1, 37)$
$J_{2,6}(35, 1, 44)$	$J_{2,7}(42, 1, 51)$	$J_{2,8}(49, 1, 58)$	$J_{2,9}(56, 1, 65)$	$J_{2,10}(63, 1, 72)$
$J_{3,1}(0, 3, 10)$	$J_{3,2}(10, 3, 20)$	$J_{3,3}(20, 3, 30)$	$J_{3,4}(30, 3, 40)$	$J_{3,5}(40, 3, 50)$
$J_{3,6}(50, 3, 60)$	$J_{3,7}(60, 3, 70)$			
$J_{4,1}(0, 7, 35)$	$J_{4,2}(35, 7, 70)$			

Tabelle 1: Jobs für Aufgabe 1

Anmerkungen:

- Die Indizes der Jobs müssen mit Komma getrennt werden, da es sowohl für T_1 als auch für T_2 mehr als 9 Jobs gibt (Verwechslungsgefahr!).
- Es sind $2+33+14=49$ Knoten (sowie 99 Kanten).

Das resultierende Netz ist in Abbildung 1 skizziert. Die Kanten von Q zu Jobs sowie von den Frames zu S wurden der Übersichtlichkeit halber weggelassen, ebenso alle Kapazitäten. Die Knoten wurden für eine spätere programmtechnische Bearbeitung des Problems bereits durchnummeriert (Q hat die Nummer 1, S die 49).

Offenbar passen alle Jobs von T_1 , T_2 , T_3 und T_4 ausgezeichnet in das Raster der 14 Frames.

- c) Zunächst ist das Netz in die DIMACS-Syntax zu transformieren, welche äußerst simpel gehalten ist. Es gibt 4 verschiedene Zeilen:

- Kommentare, die mit einem 'c' eingeleitet werden
- Die so genannte Problem-Zeile, die die Größe des Netzwerkes charakterisiert (p max KNOTEN KANTEN):

p max 49 99

- genau 2 Zeilen, die die Knotennummern für Quelle und Senke angeben:

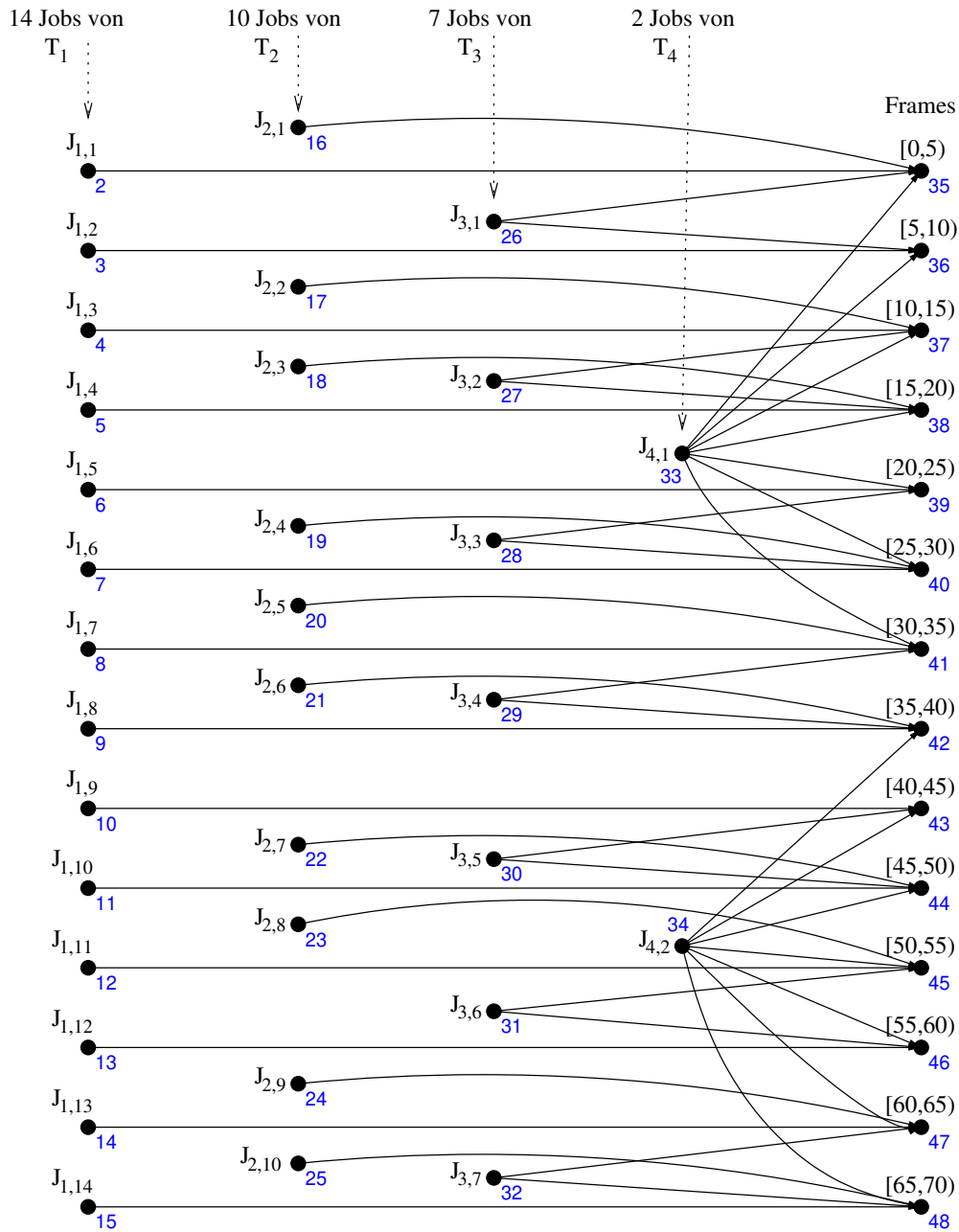


Abbildung 1: Netzwerk für Aufgabe 1b)

```
n 1 s
n 49 t
```

- pro Kante eine Zeile, die deren Quell- und Zielknoten sowie die Kapazität der Kante beschreibt (a QKNOTEN ZKNOTEN KAPAZITÄT):

```
a 1 2 1
a 1 3 1
...
a 48 49 5
```

Da es insgesamt 99 Kanten gibt, sind 99 entsprechende Zeilen notwendig. Es empfiehlt sich, sorgfältig zu arbeiten und großzügig zu kommentieren, weil das Debugging dieser Datei äußerst schwierig ausfallen dürfte.

Die resultierende Datei ist network-1.dimacs.

Nun können wir mittels PRF den maximalen Fluß bestimmen:

```
robge@ipaetz2:~/src/prf/1.9$ ./df.exe < network-1.dimacs
c
c maxflow - Dinitz algorithm
c
c nodes:          49
c arcs:           99
c
c time:           0.00
c flow:           59
c
```

PRF ermittelt einen maximalen Fluß von 59, der exakt der Summe aller Ausführungszeiten der beteiligten Jobs entspricht. Es existieren also tatsächlich gültige Schedules für die Taskmenge bei einer Framegröße von $f = 5$.

PRF übernimmt einen *Verbosity*-Parameter, der, auf 2 gesetzt, das Programm veranlasst, den tatsächlichen Fluss für jede Kante auszugeben:

```
robge@ipaetz2:~/src/prf/1.9$ ./df.exe 2 < network-1.dimacs
...
c time:           0.00
c flow:           59
c
s 59
f      1      2      1
f      1     19      1
f      1      3      1
f      1     20      1
```

...

Es ist also beim Maximalfluss von $\rho = 59$ ein tatsächlicher Fluß von Knoten 1 (der Quelle) zu den Knoten 2, 19, 3, 20 (usw.) mit Stärke 1 zu verzeichnen.

Das vollständige Resultat enthält die Datei network-1.flow

Zuguterletzt muss aus diesem Fluß der Schedule generiert werden. Dazu müssen die Flüsse, die in Frame-Knoten (Nr. 35–48) *enden*, als in diesen Frames ausgeführte Jobs interpretiert werden. Dies sind die folgenden 36 individuellen Flüsse:

Q	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Z	35	36	37	38	39	40	41	42	43	44	45	46	47	48	35	37	38	40
F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Ein Beispiel, dem der durch PRF ermittelte Fluß zugrunde liegt, zeigt Abbildung 2.

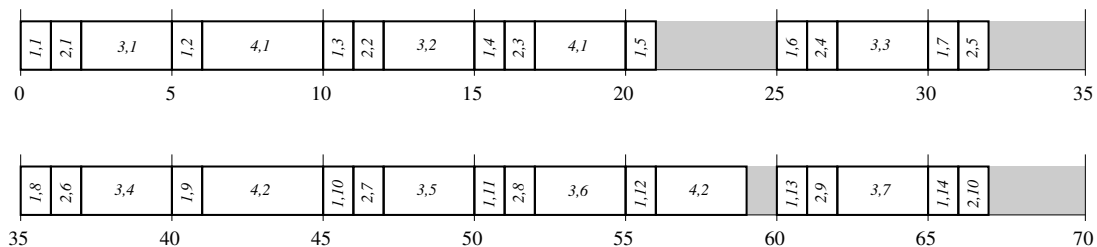


Abbildung 2: Resultierender Schedule aus Aufgabe 1c) für $f = 5$

2.
 - $H = 12$
 - Framegröße: $f \geq 2.4$, $f \in \{2, 3, 4, 6\}$, $f = 4$
 - \rightsquigarrow 3 Frames, 9 Jobs, $\sum t_e = 11.8$ (!)
 - Jobs von $T_1(t_\phi, t_e, t_d)$: $J_{11}(0, 1, 7)$, $J_{12}(3, 1, 10)$, $J_{13}(6, 1, 13)$, $J_{14}(9, 1, 16)$
 - Jobs von T_2 : $J_{21}(0, 1, 4)$, $J_{22}(4, 1, 8)$, $J_{23}(8, 1, 12)$
 - Jobs von T_3 : $J_{31}(0, 2.4, 8)$, $J_{32}(6, 2.4, 14)$
 - Abbildung 3 zeigt ein Netzwerk mit gültigem Fluß. In zwei Fällen wird jedoch die Regel verletzt, Jobs nur in den Frames abzuarbeiten, deren Beginn gleich oder später t_r des betreffenden Jobs liegt (J_{12} und J_{32}); die entsprechenden Flüsse sind gestrichelt eingezeichnet. Ohne Verletzung dieser Regel ist ein Schedule für $f = 4$ unmöglich. Der Schedule ist in Abbildung 4 dargestellt.

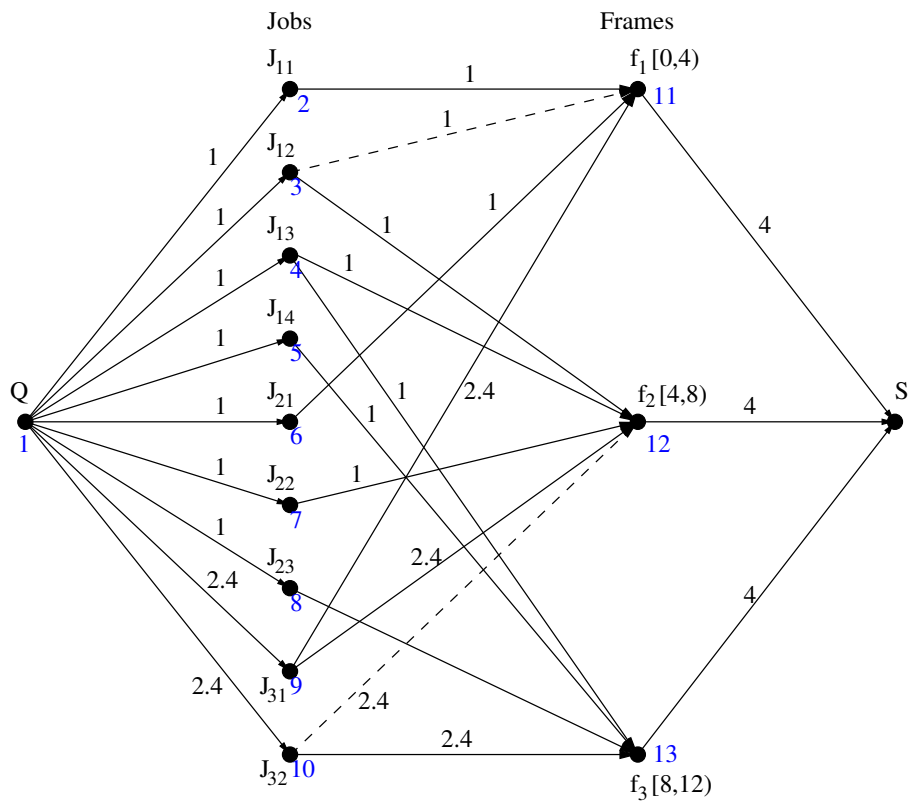


Abbildung 3: Netzwerk für $T_1(3, 1, 7)$, $T_2(4, 1)$, $T_3(6, 2.4, 8)$ sowie $f = 4$

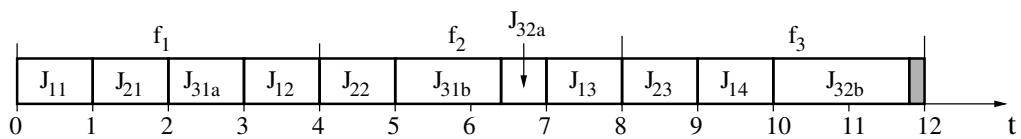


Abbildung 4: Schedule zu Aufgabe 4.